Voice over IP: Risks, Threats and Vulnerabilities

Angelos D. Keromytis
Symantec Research Labs Europe
Sophia-Antipolis, France

***Abstract*— Voice over IP (VoIP) and Internet Multimedia Subsystem (IMS) technologies are rapidly being adopted by consumers, enterprises, governments and militaries. These technologies offer higher flexibility and more features than traditional telephony (PSTN) infrastructures, as well as the potential for lower cost through equipment consolidation and, for the consumer market, new business models. However, VoIP/IMS systems also represent a higher complexity in terms of architecture, protocols and implementation, with a corresponding increase in the potential for misuse. Here, we begin to examine the current state of affairs on VoIP/IMS security through a survey of known/disclosed security vulnerabilities in bug-tracking databases. This paper should serve as a starting point for understanding the threats and risks in a rapidly evolving set of technologies that are seeing increasing deployment and use. Our goal is to gain a better understanding of the security landscape with respect to VoIP/IMS, toward directing future research in this and other similar emerging technologies.**

I. Introduction

The rate at which new technologies are being introduced and adopted by society has been steadily accelerating throughout human history. The advent of pervasive computing and telecommunications has reinforced this trend. In this environment of constant innovation, individuals, governments and organizations have been struggling to manage the tension between reaping the benefits of new technologies while understanding and managing their risks. In this struggle, cost reductions, convenience and new features typically overcome security concerns. As a result, security experts (but also the government and the courts of law) are often left with the task of playing "catch up" with those who exploit flaws to further their own goals. This is the situation we find ourselves in with respect to one popular class of technologies, collectively referred to as Voice over IP (VoIP).

VoIP, sometimes also referred to as Internet Multimedia Subsystem (IMS), refers to a class of products that enable advanced communication services over data networks. While voice is a key aspect in such products, video and other capabilities (*e.g.*, collaborative editing and whiteboard sharing, file sharing, calendaring) are supported. The key advantages of VoIP/IMS are flexibility and low cost. The former derives from the (generally) open architectures and software-based implementation, while the latter is due to new business models, equipment and network-link consolidation, and ubiquitous consumer-grade broadband connectivity.

Due to these benefits, VoIP has seen rapid uptake in both the enterprise and consumer markets. An increasing number of enterprises are replacing their internal phone switches with VoIP-based implementations, both to introduce new features and to eliminate redundant equipment. Consumers have embraced a slew of technologies with different features and costs, including P2P calling, Internet-to-phone network bridging, and wireless VoIP. These new technologies and business models are being promoted by a new generation of startup companies that are challenging the traditional status quo in telephony and personal telecommunications. As a result, a number of PSTN providers have already completed or are in the process of transitioning from circuit-switched networks to VoIP-friendly packet-switched backbones. Finally, as the commercial and consumer sectors go, so do governments and militaries due to cost reduction concerns and the general dependence on Commercial Off The Shelf (COTS) equipment for the majority of their computing needs.

However, higher complexity is often the price we pay for more flexibility. In the case of VoIP/IMS technologies, a number of factors contribute to architectural, protocol, implementation and operational complexity:

- The number and complexity of the various features integrated in a product are perhaps the single largest source of complexity. For example, voice and video transmission typically allow for a variety of codecs which may be used in almost-arbitrary combinations. Since one of the biggest selling points for VoIP/IMS is feature-richness and the desire to unify personal communications under the same umbrella, this is a particularly pertinent concern.
- Openness and modularity, generally considered desirable traits, allow for a number of independent implementations and products. Each of these comes with its own parameters and design choices. Interoperability concerns and customer feedback then lead to an ever-growing baseline of supported features for all products. A compounding factor to increasing complexity for many of the open VoIP protocols is the "design-by-committee" syndrome, which typically leads to larger, more inclusive specifications than would otherwise be the case (*e.g.*, in a closed, proprietary environment such as the wireline telephony network from 20 years ago).
- Because VoIP systems are envisioned to operate in a variety of environments, business settings, and network conditions, they must offer considerable configurability, which in turns leads to high complexity. Of particular concern are unforeseen feature interactions and other emergent properties.
- Finally, VoIP are generally meant to work over a public data network (*e.g.*, the Internet), or an enterprise/operator

network that uses the same underlying technology. As a result, there is a substantial amount of (strictly speaking) non-VoIP infrastructure that is critical for the correct operation of the system, including such protocols/services as DHCP [1], DNS [2], [3], TFTP/BOOTP [4], [5], NAT [6] (and NAT traversal protocols such as STUN [7]), NTP [8], SNMP [9], routing, the web (HTTP [10], [11], TLS/SSL [12], *etc.*) and many others. As we shall see, even a "perfectly secure" VoIP system can be compromised by subverting elements of this infrastructure.

Because of this complexity, which manifests itself both in terms of configuration options and size of the code base for VoIP implementations, VoIP systems represent a very large attack surface. Thus, one should expect to encounter, over time, security problems arising from design flaws (*e.g.*, exploitable protocol weaknesses), undesirable feature interactions (*e.g.*, combinations of components that make new attacks possible or existing/known attacks easier), unforeseen dependencies (*e.g.*, compromise paths through seemingly unrelated protocols), weak configurations, and, not least, implementation flaws.

In this paper, we attempt a first effort at mapping out the space of VoIP threats and risks by conducting a survey of the "actually seen" vulnerabilities and attacks, as reported by the popular press and by bug-tracking databases. Our work is by necessity of evolutionary nature, and this paper represents a current (and limited) snapshot of the complete space. Nonetheless, we believe that it will serve as a valuable starting point for understanding the bigger problem, and as a basis for a more comprehensive analysis in the future.

Paper Organization: The remainder of this paper is organized as follows. Section II contains a brief overview of two major VoIP technologies, SIP and UMA. While we refer to other VoIP/IMS systems throughout the discussion, we focus on the specific two technologies as they are both representative, widely used, and well-documented. We discuss VoIP threats in Section III, placing known attacks against VoIP systems within the taxonomy proposed by the VoIP Security Alliance[1]. We analyze our findings in Section IV, and conclude with some preliminary thoughts on the current state of VoIP security, and on possible future directions for security research and practices in Section V.

II. VoIP Technologies Overview

In their simplest form, Voice over IP protocols simply enable two (or more) devices to transmit and receive real-time audio traffic that allows their respective users to communicate. In general, VoIP architectures are partitioned in two main components: signaling and media transfer. Signaling covers both abstract notions, such as endpoint naming and addressing, and concrete protocol functions such as parameter negotiation, access control, billing, proxying, and NAT traversal. Depending on the architecture, quality of service (QoS) and device configuration/management may also be part of the signaling protocol (or protocol family). The media transfer aspect of VoIP systems generally includes a comparatively simpler protocol for encapsulating data, with support for multiple codecs and (often, but not always) content security. A commonly used media transfer protocol is RTP [13], with a version supporting encryption and integrity (SRTP [14]) defined but not yet widely used. The RTP protocol family also includes RTCP, which is used to control certain RTP parameters between communicating endpoints.

However, a variety of other features are generally also desired by users and offered by providers as a means for differentiation by competing technologies and services, such video, integration with calendaring and file sharing, and bridging to other networks (*e.g.*, to the "regular" telephony network). Furthermore, a number of different decisions may be made when designing a VoIP system, reflecting different requirements and approaches to addressing, billing, mobility, security and access control, usability, and other issues. Consequently, there exist a variety of different VoIP/IMS protocols and architectures. For concreteness, we will focus our attention on a popular and widely deployed technology: the Session Initiation Protocol (SIP) [15]. We will also discuss the Unlicensed Mobile Access (UMA) architecture [16], as a different approach to VoIP that is gaining traction among wireless telephony operators. In the rest of this section, we give a high-level overview of SIP and UMA, followed by a brief description of the salient points of a few other popular VoIP systems, such as H.323 and Skype. We will refer back to this overview when discussing the threat space and specific vulnerabilities in Section III.

A. Session Initiation Protocol

SIP is a protocol standardized by the Internet Engineering Task Force (IETF), and is designed to support the setup of bidirectional communication sessions including, but not limited to, VoIP calls. It is similar in some ways to HTTP, in that it is text-based, has a request-response structure, and even uses a mechanism based on the HTTP Digest Authentication [17] for user authentication. However, it is an inherently stateful protocol that supports interaction with multiple network components (*e.g.*, middleboxes such as PSTN bridges). While its finite state machine is seemingly simple, in practice it has become quite large and complicated — an observation supported by the fact that the main SIP RFC [15] is one of the longest ever defined.

SIP can operate over a number of transport protocols, including TCP [18], UDP [19] and SCTP [20]. UDP is generally the preferred method due to simplicity and performance, although TCP has the advantage of supporting TLS protection of call setup. However, recent work on Datagram TLS (DTLS) [21] may render this irrelevant. SCTP, on the other hand, offers several advantages over both TCP and UDP, including DoS resistance [22], multi-homing and mobility support, and logical connection multiplexing over a single channel.

In the SIP architecture, the main entities are end points (whether softphones or physical devices), a proxy server, a registrar, a redirect server, and a location server. Figure 1 shows a high-level view of the SIP entity interactions. The

[1] `http://www.voipsa.org/`

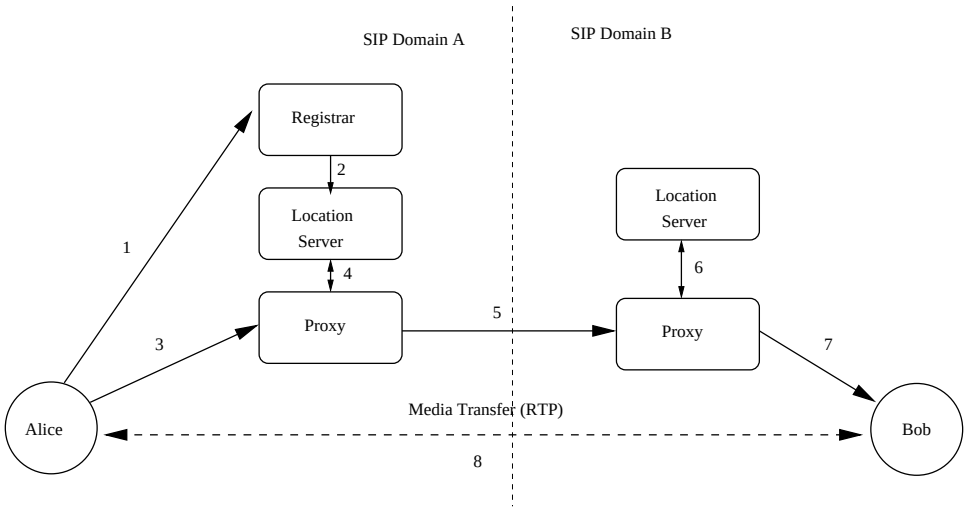

Fig. 1. Session Initiation Protocol (SIP) entity interactions. User Alice registers with her domain's Registrar (1), which stores the information in the Location Server (2). When placing a call, Alice contacts her local Proxy Server (3), which may consult the Location Server (4). A call may be forwarded to another Proxy Server (5), which will consult its domain Location Server (6) before forwarding the call to the final recipient. After the SIP negotiation terminates, RTP is used directly between Alice and Bob to transfer media content. For simplicity, this diagram does not show the possible interaction between Alice and a Redirection Server (which would, in turn, interact with the Location Server).

registrar, proxy and redirect servers may be combined, or they may be separate entities operated independently. Endpoints communicate with a registrar to indicate their presence. This information is stored in the location server. A user may be registered via multiple endpoints simultaneously.

During call setup, the endpoint communicates with the proxy which uses the location server to determine where the call should be routed to. This may be another endpoint in the same network (*e.g.*, within the same enterprise), or another proxy server in another network. Alternatively, endpoints may use a redirect server to directly determine where a call should be directed to; redirect servers consult with the location server in the same way that proxy servers operate during call setup. Once an end-to-end channel has been established (through one or more proxies) between the two endpoints, SIP negotiates the actual session parameters (such as the codecs, RTP ports, *etc.*) using the Session Description Protocol (SDP) [23].

Figure 2 shows the message exchanges during a two-party call setup. Alice sends an INVITE message to the proxy server, optionally containing session parameter information encoded within SDP. The proxy forwards this message directly to Bob, if Alice and Bob are users of the same domain. If Bob is registered in a different domain, the message will be relayed to Bob's proxy, and from there to Bob. Note that the message may be forwarded to multiple endpoints, if bob is registered from multiple locations. While these are ringing (or otherwise indicating that a call setup is being requested), RINGING messages are sent back to Alice. Once the call has been accepted, an OK message is sent to Alice, containing his preferred parameters encoded within SDP. Alice responds with an ACK message. Alice's session parameter preferences may be encoded in the INVITE or the ACK message.

Following this exchange, the two endpoints can begin transmitting voice, video or other content (as negotiated) using the agreed-upon media transport protocol, typically RTP. While the signaling traffic may be relayed through a number of SIP proxies, the media traffic is exchanged directly between the two endpoints. When bridging different networks, *e.g.*, PSTN and SIP, media gateways may disrupt the end-to-end nature of the media transfer. These entities translate content (*e.g.*, audio) between the formats that are supported by the different networks.

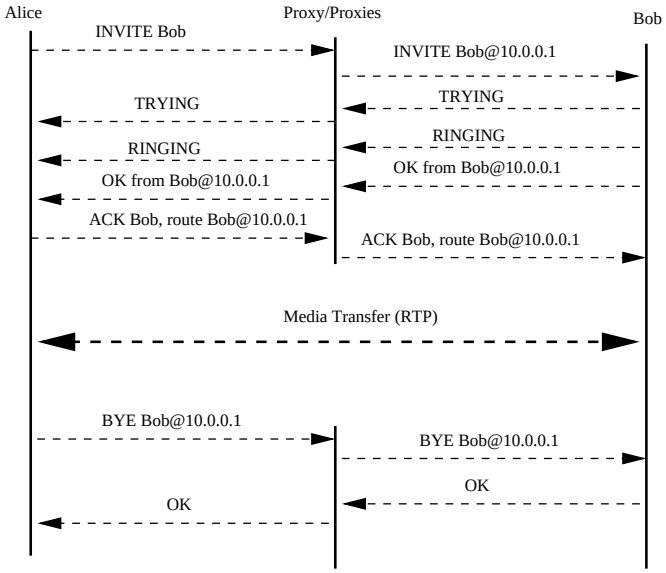

Fig. 2. Message exchanges during a SIP-based two-party call setup.

Because signaling and media transfer operate independent of each other, the endpoints are responsible for indicating to

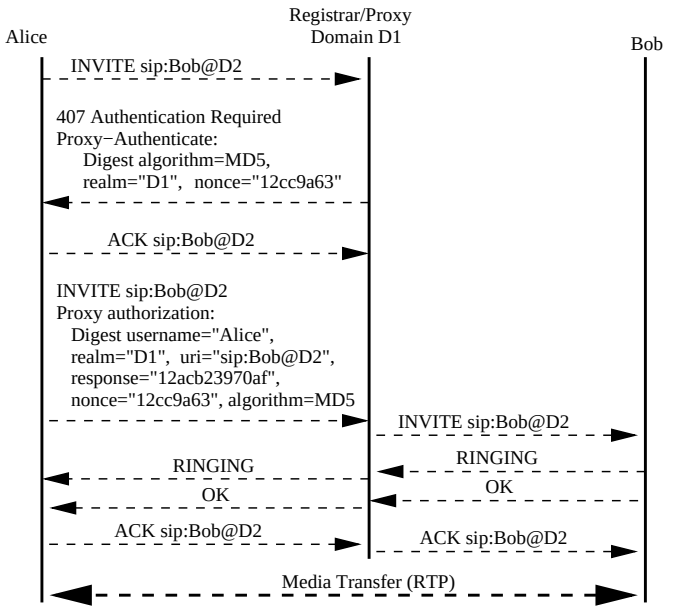

Fig. 3. SIP Digest Authentication

the proxies that the call has been terminated, using a BYE message which is relayed through the proxies along the same path as the call setup messages.

There are many other protocol interactions supported by SIP, that cover many common (and uncommon) scenarios including call forwarding (manual or automatic), conference calling, voicemail, *etc.* Typically, this is done by semantically overloading SIP messages such that they can play various roles in different parts of the call. We shall see in Section III examples of how this flexibility and protocol modularity can be used to attack the system.

All SIP traffic is transmitted over port 5060 (UDP or TCP). The ports used for the media traffic, however, are dynamic and negotiated via SDP during call setup. This poses some problems when Network Address Translation (NAT) or firewalls are traversed. Typically, these have to be stateful and understand the SIP exchanges so that they can open the appropriate RTP ports for the media transfer. In the case of NAT traversal, endpoints may use protocols like STUN to enable communication. Alternatively, the Universal Plug-and-Play (uPnP) protocol [2] may be used in some environments, such as residential broadband networks consisting of a single subnet behind a NAT gateway.

Authentication between endpoints, the registrar and the proxy typically uses HTTP Digest Authentication, as shown in Figure 3. This is a simple challenge-response protocol that uses a shared secret key along with a username, domain name, a nonce, and specific fields from the SIP message to compute a cryptographic hash. Using this mechanism, passwords are not transmitted in plaintext form over the network. It is worth noting that authentication may be requested at almost any point

[2] `http://www.upnp.org/`

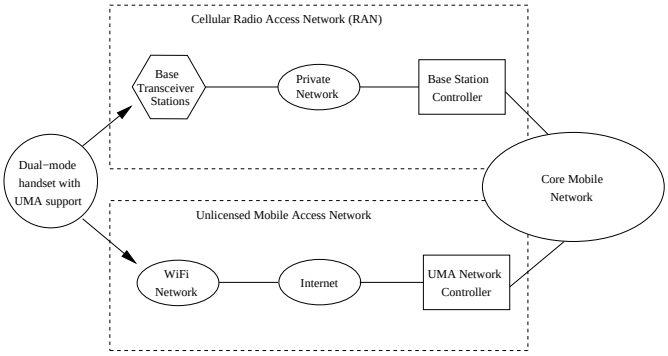

Fig. 4. Unlicensed Mobile Access (UMA) conceptual architecture

during a call setup. We shall later see an example where this can be abused by a malicious party to conduct toll fraud in some environments.

For more complex authentication scenarios, SIP can use S/MIME encapsulation [24] to carry complex payloads, including public keys and certificates. When TCP is used as the transport protocol for SIP, TLS can be used to protect the SIP messages. TLS is required for communication among proxies, registrars and redirect servers, but only recommended between endpoints and proxies or registrars. Alternatively, IPsec [25] may be used to protect all communications, regardless of the transport protocol. However, because few implementations integrate SIP, RTP and IPsec, it is left to system administrators to figure out how to setup and manage such configurations.

B. Unlicensed Mobile Access

UMA is a 3GPP standard for enabling transparent access to mobile circuit-switched voice networks, packet-switch data networks and IMS services using any IP-based substrate. Handsets supporting UMA can roam between the operator's wireless network (usually referred to as a Radio Access Network, or RAN) and the Internet without losing access. For example, a call that is initiated over the RAN can then be routed, without being dropped and with no user intervention, over the public Internet if conditions are more favorable (*e.g.*, stronger WiFi signal in the user's premises, or in a hotel wireless hotspot while traveling abroad). For consumers, UMA offers better connectivity and the possibility of lower cost by enabling new business models and reducing roaming charges (under some scenarios). For operators, UMA reduces the need for additional spectrum, cellphone towers and related equipment. A variety of cellphones supporting UMA over WiFi currently exist, along with home gateways and USB-stick softphones. More recently, some operators have introduced femtocells (ultra-low power RAN cells intended for consumer-directed deployment) that can act as UMA gateways, allowing any mobile handset to take advantage of UMA where such devices are deployed.

The basic approach behind UMA is to encapsulate complete GSM and 3G radio frames (except for the over-the-air crypto) inside IP packets. These can then be transmitted over any IP

network, including the Internet. This means that the mobile operator can continue to use the existing back-end equipment; all that is needed is a gateway that decapsulates the GSM/3G frames and injects them to the existing circuit-switched network (for voice calls), as can be seen in Figure 4.

To protect both signaling and media traffic confidentiality and integrity while traversing untrusted (and untrustworthy) networks, UMA uses IPsec. All traffic between the handset (or, more generally, UMA endpoint) and the provider's UMA Network Controller (or a firewall/VPN concentrator screening traffic) is encrypted and integrity-protected using ESP [26]. The use of IPsec provides a high level of security for the traffic, once keys and other parameters have been negotiated. For that purpose, the IKEv2 key management protocol [27] is used. Authentication uses the EAP-SIM [28] (for GSM handsets) and EAP-AKA [29] (for UMTS handsets) profiles. Authentication is asymmetric: the provider authenticates to the handset using digital signatures and public key certificates, while the handset authenticates using a SIM-embedded secret key. It is worth pointing out that UMA provides stronger authentication guarantees than the baseline cellphone network, in that the provider does not authenticate to the handset in a RAN. Furthermore, the cryptographic algorithms used in IPsec (AES and 3DES) are considered significantly stronger than the on-the-air algorithms used in GSM.

Despite the use of strong cryptography and sound protocols, UMA introduces some new risks in the operator networks, since these now have to be connected to the public Internet in a much more intimate fashion. In particular, the security gateway must process IPsec traffic, including the relatively complex IKEv2 protocol, and a number of UMA-related discovery and configuration protocols. These increase the attack surface and overall security exposure of the operators significantly.

C. Other VoIP/IMS Systems

H.323 is an ITU-defined protocol family for VoIP (audio and video) over packet-switched data networks. The various subprotocols are encoded in ASN.1 format. In the H.323 world, the main entities are terminals (software or physical phones), a gateway, a gatekeeper and a back-end service. The gatekeeper is responsible for address resolution, controlling bandwidth use and other management functions, while the gateway connects the H.323 network with other networks (*e.g.*, PSTN, or a SIP network). The back-end service maintains data about the terminals, including configuration, access and billing rights, *etc.* An optional multipoint control unit may also exist to enable multipoint communications, such as a teleconference. To setup a H.323 call, terminals first interact with the gatekeeper using the H.225 protocol over either TCP or UDP to receive authorization and perform address resolution. Using the same protocol, they then establish the end-to-end connection to the remote terminal (possibly through one or more gateways). At that point, H.245 over TCP is used to negotiate the parameters for the actual media transfer, including ports, which uses RTP (as in the case of SIP). A number of other protocols within the H.323 framework covering security, interoperability with PSTN, teleconferencing, and others. Authentication may be requested at several steps during call setup, and typically depends on symmetric keys but may also use digital signatures. Voice encryption is also supported through SRTP and MIKEY [30]. Unlike SIP, H.323 does not use a well-known port, making firewall traversal even more complicated.

Skype[3] is a peer-to-peer VoIP system that was originally available as a softphone for desktop computers but has since been integrated into cellphones and other handheld devices, either as an add-on or as the exclusive communication mechanism. It offers voice, video, and text messaging to all other Skype users free of charge, and provides bridging (typically for a fee) to the PSTN both for outgoing and incoming calls and text messages (SMS). The underlying protocol is proprietary, and the software itself incorporates several anti-reverse engineering techniques. Nonetheless, some analysis [31], [32] and reverse engineering [33] have taken place, indicating both the ubiquitous use of strong cryptography and the presence of some software bugs (at the time of the work). The system uses a centralized login server but is otherwise fully distributed with respect to intra-Skype communications.

A number of chat (IM) networks, such as the AOL Instant Messenger, Microsoft's Live Messenger, Yahoo! Messenger, and Google Talk offer voice and video capabilities as well. Although each network uses its own (often proprietary) protocol, there exist bridges between most of them, allowing inter-IM communication at the text level. In most of these networks, users can place outgoing voice calls to the PSTN. Some popular IM clients also integrate SIP support.

III. VoIP Threats

In trying to understand the threat space against VoIP, our approach is to place known vulnerabilities within a structured framework. While a single taxonomy is not likely to be definitive, using several different viewpoints and mapping the vulnerability space along several axis may reveal trends and areas that merit further analysis.

As a starting point, we use the taxonomy provided by the Voice over IP Security Alliance (VoIPSA)[4]. VoIPSA is a vendor-neutral, not for profit organization composed of VoIP and security vendors, organizations and individuals with an interest in securing VoIP protocols, products and installations. In addition, we place the surveyed vulnerabilities within the traditional threat space of confidentiality, integrity, availability (CIA). Finally, we consider whether the vulnerabilities exploit bugs in the protocol, implementation or system configuration. In future work, we hope to expand the number of views to the surveyed vulnerabilities and to provide more in-depth analysis.

The VoIPSA security threat taxonomy [34] aims to define the security threats against VoIP deployments, services, and end users. The key elements of this taxonomy are:

[3] `http://www.skype.com/`
[4] `http://www.voipsa.org/`

1) **Social threats** are aimed directly against humans. For example, misconfigurations, bugs or bad protocol interactions in VoIP systems may enable or facilitate attacks that misrepresent the identity of malicious parties to users. Such attacks may then act as stepping stones to further attacks such as phishing, theft of service, or unwanted contact (spam).
2) **Eavesdropping, interception, and modification threats** cover situations where an adversary can unlawfully and without authorization from the parties concerned listen in on the signaling (call setup) or the content of a VoIP session, and possibly modify aspects of that session while avoiding detection. Examples of such attacks include call re-routing and interception of unencrypted RTP sessions.
3) **Denial of service threats** have the potential to deny users access to VoIP services. This may be particularly problematic in the case of emergencies, or when a DoS attack affects all of a user's or organization's communication capabilities (*i.e.,* when all VoIP and data communications are multiplexed over the same network which can be targeted through a DoS attack). Such attacks may be VoIP-specific (exploiting flaws in the call setup or the implementation of services), or VoIP-agnostic (*e.g.,* generic traffic flooding attacks). They may also involve attacks with physical components (*e.g.,* physically disconnecting or severing a cable) or through computing or other infrastructures (*e.g.,* disabling the DNS server, or shutting down power).
4) **Service abuse threats** covers the improper use of VoIP services, especially (but not exclusively) in those situations where such services are offered in a commercial setting. Examples of such threats include toll fraud and billing avoidance [35], [36].
5) **Physical access threats** refer to inappropriate/unauthorized physical access to VoIP equipment, or to the physical layer of the network (following the ISO 7-layer network stack model).
6) **Interruption of services threats** refer to non-intentional problems that may nonetheless cause VoIP services to become unusable or inaccessible. Examples of such threats include loss of power due to inclement weather, resource exhaustion due to over-subscription, and performance issues that degrade call quality.

In our discussion of vulnerabilities (whether theoretical or demonstrated) that follows, we shall mark each item with a tuple (V, T, K), where:

- $V \in \{1, 2, 3, 4, 5, 6\}$, where each number refers to an element in the VoIPSA threat taxonomy from above
- $T \in \{C_1, I_1, A_1\}$, referring to Confidentiality, Integrity and Availability respectively
- $K \in \{P_2, I_2, C_2\}$, referring to Protocol, Implementation and Configuration respectively

For example, an item marked as $(1, C_1, C_2)$ refers to a vulnerability that targets the user (Social threat), violating Confidentiality via a Configuration problem or bug. In some cases, the same underlying vulnerability may be used to perform different types of attacks. We will be discussing all such significant attack variants.

A. Disclosed Vulnerabilities

Threats against VoIP system availability by exploiting implementation weaknesses are fairly common. For example, some implementations where shown to be vulnerable to crashes or hanging (livelock) when given empty, malformed, or large volumes of [37]–[150] INVITE or other messages $(3, A_1, I_2)$. It is worth noting that the same vulnerability may be present across similar protocols on the same platform and product [44] due to code sharing and internal software structure, or to systems that need to understand VoIP protocols but are not nominally part of a VoIP system [151]. The reason for the disproportionately large number of denial of service vulnerabilities is because of the ease with which such failure can be diagnosed, especially when the bug is discovered through automated testing tools (*e.g.,* fuzzers). Many of these vulnerabilities may in fact be more serious than a simple denial of service due to a crash, and could possibly lead to remote code injection and execution.

Unexpected interactions between different technologies used in VoIP systems can also lead to vulnerabilities. For example, in some cases cross-site scripting (XSS) attacks were demonstrated against the administrator- and customer-facing management interface (which was web-based) by injecting malicious Javascript in selected SIP messages [152]–[159] $(1, I_1, I_2)$, often through SQL injection vulnerabilities [160], [161]. The same vulnerability could also be used to commit toll fraud by targeting the underlying database $(4, I_1, I_2)$. XSS attacks that are not web-oriented have also been demonstrated, with one of the oldest VoIP-related vulnerabilities [162] permitting shell command execution $(1, I_1, I_2)$. Another web-oriented attack vector is Cross Site Request Forgery (CSRF), whereby users visiting a malicious page can be induced to automatically (without user intervention, and often without any observable indications) perform some action on the web servers (in this case, VoIP web-based management interface) that their browser is already authenticated to [163] $(1, I_1, I_2)$. Other privilege-escalation vulnerabilities through the web interface also exist [164] $(1, I_1, I_2)$.

The complexity of the SIP finite state machine has sometimes led to poor implementations. For example, one vulnerability [165] allowed attackers to confuse a phone receiving a call into silently completing the call, which allowed the adversary to eavesdrop on the device's surroundings $(2, C_1, I_2)$. The same vulnerability could be used to deny call reception at the target, since the device was already marked as busy $(3, A_1, I_2)$. In other cases, it is unclear to developers what use of a specific protocol field may be, in which case they may silently ignore it. Occasionally, such information is critical for the security of the protocol exchange, and omitting or not checking it allows adversaries to perform attacks such man-in-the-middle or traffic interception [166] $(2, C_1 + I_1, I_2)$, or

to bypass authentication checks [167], [168] $(4, I_1, I_2)$.

Since SIP devices are primarily software-driven, they are vulnerable to the same classes of vulnerabilities as other software. For example, buffer overflows are possible even against SIP "hardphones", much less softphones, allowing adversaries to gain complete control of the device [40], [64], [65], [169]–[191] $(2, I_1, I_2)$. Such vulnerabilities typically arise from a combination of poor (non-defensive) programming practices, insufficient testing, and the use of languages, such as C and C++ that support unsafe operations. Sometimes, these vulnerabilities appear in software that is not directly used in VoIP but must be VoIP-aware, *e.g.,* firewalls [192] or protocol analyzers [193] $(2, I_1, I_2)$. It is also worth noting that these are not the only types of vulnerabilities that can lead to remote code execution [59], [194]–[198]. Other input validation failures can allow attackers to download arbitrary files from a user's machine $(1, C_1, I_2)$ or to place calls [199] $(1, I_1, I_2)$ by supplying specially encoded URIs [200] or other parameters.

A significant risk with VoIP devices is the ability of adversaries to misrepresent their identity (*e.g.,* their calling number). Such vulnerabilities [201] sometimes arise due to the lack of cross-checking of information provided across several messages during call setup and throughout the session $(1, I_1, I_2)$.

Similar failures to cross-check and validate information can lead to other attacks, such as indicating whether there is pending voicemail for the user [202] $(1, I_1, I_2)$, or where attackers may spoof incoming calls by directly connecting to a VoIP phone [203]–[205] $(1, I_1, I_2)$.

Undocumented, on-by-default features are another source of vulnerabilities. These are often remnants from testing and debugging during development that were not disabled when a product shipped [206]–[210]. As a result, they often offer privileged access to services and data on a device that would not be otherwise available [211]–[216] $(1, C_1, I_2)$. One particularly interesting vulnerability allowed an attacker to place outgoing calls through the web management interface [217], [218] $(4, I_1, C_2)$.

A significant class of vulnerabilities in VoIP devices revolves around default configurations, and in particular default usernames and passwords [216], [219]–[228] $(2, C_1 + I_1, C_2)$. Lists of default accounts are easy to find on the Internet via a search engine. Users often do not change these settings; ironically, this seems to be particularly so for administrative accounts, which are rarely (if ever) used in the home/SOHO environment. Other default settings involve NTP servers [229] and DNS servers [230] $(2, C_1 + I_1, C_2)$.

Call interception vulnerabilities are a big concern with VoIP systems, given the plethora of tools for decoding video and audio streams and the ease of eavesdropping network traffic, especially on the local subnet. Sometimes, such vulnerabilities arise from strange protocol interactions and implementation decisions. For example, caching the location (address) of a VoIP phone based on the IP address used during boot time (using TFTP) seems a reasonable approach; however, since the boot and VoIP stacks are not necessarily tightly integrated, interaction with one protocol can have adverse effects (*e.g.,* changing the perceived location of the phone) in the other protocol [231] $(2, C_1, I_2)$. Other instances of such vulnerabilities involve improper/insufficient credential checking by the registrar or proxy [232] or by the SNMP server [233], which can lead to traffic interception $(2, C_1, I_2)$ and user impersonation $(1, I_1, I_2)$.

The integration of several capabilities in VoIP products, *e.g.,* a web server used for the management interface, can lead to vulnerabilities being imported to the VoIP environment that would not otherwise apply. In the specific example of an integrated web server, directory traversal bugs [234] or similar problems (such as lack of proper authentication in the web interface) [235], [236] can allow adversaries to read arbitrary files or other information from the device $(1, C_1, I_2)$. SIP (or, more generally, VoIP) components integrated with firewalls may also interact in undesirable ways. For example, improper handling of registration requests may allow attackers to receive messages intended for other users [237] $(2, C_1, I_2)$. Other such examples include failure to authenticate server certificates in wireless environments, enabling man-in-the-middle and eavesdropping attacks [238], [239] $(2, C_1, I_2)$.

Predictability and lack of proper use (or sources) of randomness is another vulnerability seen in VoIP products. For example, predictable values in SIP header messages [240] allows malicious users to avoid registering but continue using the service $(4, I_1, I_2)$.

Protocol responses to carefully crafted messages can reveal information about the system or its users to an attacker. Although this has been long understood in limited-domain protocols (*e.g.,* remote login), with measures taken to normalize responses such that no information is leaked, the complexity of VoIP (and other) protocols make this infeasible. As a result, information disclosure vulnerabilities abound [241], [242] $(1, C_1, I_2)$.

Some of the most serious non-implementation type of vulnerabilities are those where the specification permits behavior that is exploitable. For example, certain vendors permit the actual URI in a SIP INVITE call and the URI used as part of the Digest Authentication to differ, which (while arguably permitted by the specification) allows credential reuse and toll fraud [243], [244] $(4, I_1, P_2)$.

While rare, protocol-level vulnerabilities also exist. These represent either outright bugs in the specification, or unforeseen interaction between different protocols or protocol components. For large, complicated protocols such as SIP and H.323, where components (code, messages, *etc.*) are semantically overloaded and reused, it is perhaps not surprising that such emergent properties exist. One good example is the relay attack possible with the SIP Digest Authentication [245], whereby an adversary can reuse another party's credentials to obtain unauthorized access to SIP or PSTN services (such as calling a premium or international phone line) $(4, I_1, P_2)$. This attack, depicted in Figure 5, is possible because authentication may be requested in response to an INVITE message at any

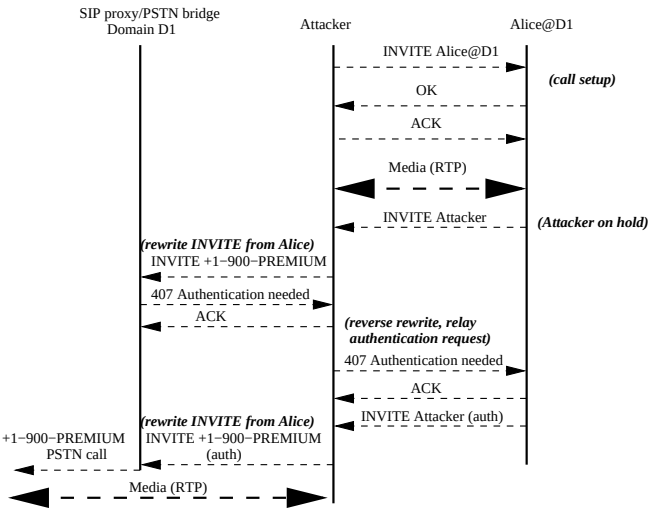

Fig. 5. SIP relay attack

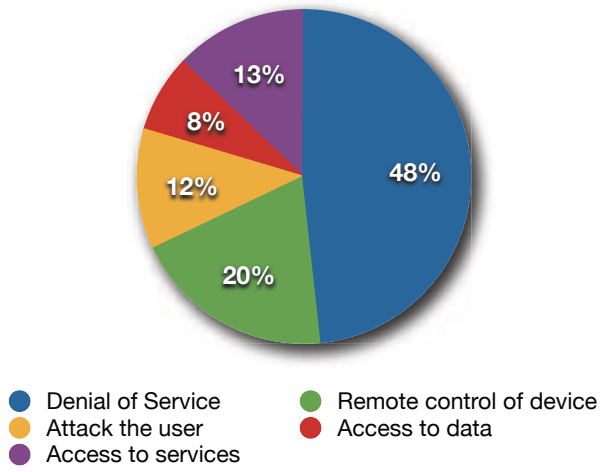

Fig. 6. Vulnerability breakdown based on effect. Most categories are self-explanatory; "attack the user" refers to vulnerabilities that permit attackers to affect the user/administrator of a device, without necessarily compromising the system or getting access to its data or services. XSS attacks and traffic eavesdropping attacks fall in this category, whereas attacks that compromise state (data) resident on the system fall in the "access to data" category.

time during a call, and the responder may issue an INVITE message during a call either automatically (because of timer expirations) or through a user action (*e.g.*, placing the caller on hold in order to do a call transfer).

IV. Discussion

Looking at the vulnerabilities we have considered, a few patterns emerge. First, as we can see in our informal classification of vulnerability effects show in Figure 6, half of the problems lead to a denial of service in either an end-device (phone, softphone) or a server (proxy, registrar, *etc.*). This is not altogether surprising, since denial of service (especially a crash) is something that is easily diagnosed. In many cases, the problem was discovered by automated testing, such as protocol or software fuzzing; software failures are relatively easy to determine in such settings. Some of these vulnerabilities could in fact turn out to be more serious, *e.g.*, a memory corruption leading to a crash could be exploitable to mount a code injection attack. The second largest class of vulnerabilities allows an adversary to control the device, whether by code injection, default passwords and services, or authentication failures. Note that we counted a few of the vulnerabilities (approximately 10%) more than once in this classification.

The same pattern with respect to the predominance of denial of service vulnerabilities holds when we look at the breakdown according to the VoIPSA taxonomy, shown in Figure 7. It should not be surprising that, given the nature of the vulnerabilities disclosed in CVE, we have no data on physical access and (accidental) interruption of services vulnerabilities. Furthermore, while "Access to Services" was a non-negligible component in the previous breakdown, it represents only 4% here. The reason for this apparent discrepancy is in the different definitions of service: the specific element in the VoIPSA taxonomy refers to VoIP-specific abuse, whereas our informal definition covers lower-level system components which may not be usable in, for example, placing fraudulent calls. One

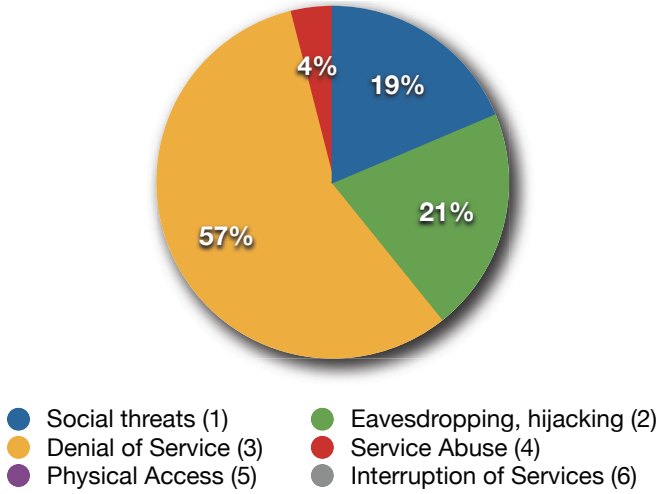

Fig. 7. Vulnerability breakdown based on VoIPSA taxonomy

other observation here is that, while the VoIPSA taxonomy covers a broad spectrum of concerns for VoIP system designers and operators, its categories are too perhaps too broad (and, in some cases, imprecise) to help with characterizing the types of bugs we have examined.

The vulnerability breakdown according to the traditional (Confidentiality, Integrity, Availability) security concerns again reflects the predominance of denial of service threats against VoIP systems, as seen in Figure 8. However, we can see that Integrity violations (*e.g.*, system compromise) are a sizable component of the threat space, while Confidentiality violations are seen in only 15% of disclosed vulnerabilities. This represents an inversion of the perceived threats by users and admin-

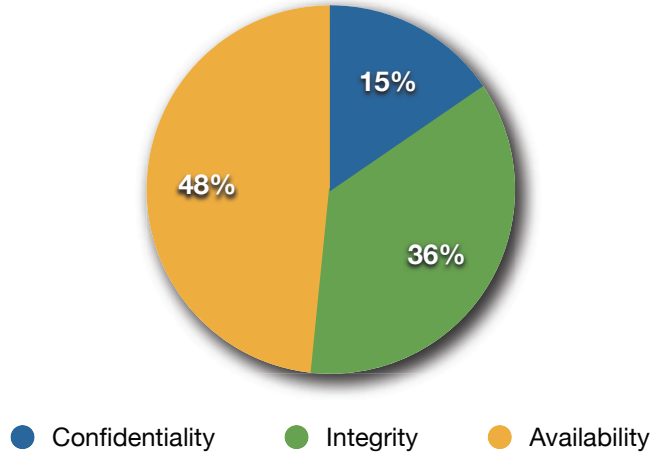

Fig. 8. Vulnerability breakdown based on "traditional" security classification (C_1, I_1, A_1)

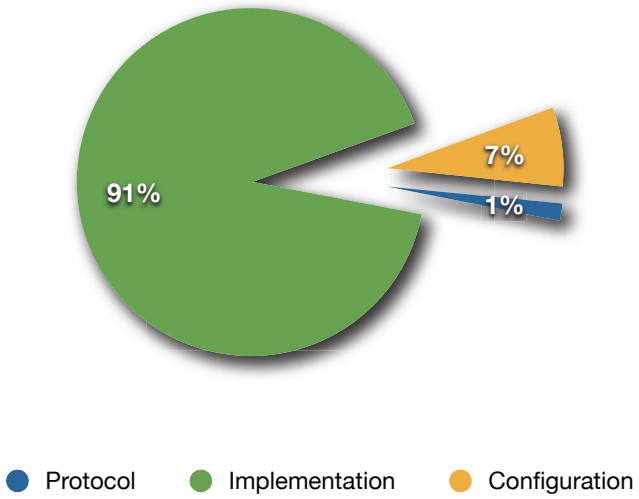

Fig. 9. Vulnerability breakdown based on source (I_2, C_2, P_2)

istrators, who (anecdotal evidence suggests) typically worry about such issues as call interception and eavesdropping.

Finally, Figure 9 shows the breakdown based on source of vulnerability. The overwhelming majority of reported problems arise from implementation issues, which should not be surprising given the nature of bug disclosure. Problems arising from configuration represented 7% of the total space, including such items as privileged services left on and default username/passwords. However, note that the true picture (*i.e.*, what actually happens with deployed systems) is probably different in that configuration problems are most likely undercounted: such problems are often site-specific and are not reported to bug-disclosure databases when discovered. On the other hand, implementation and protocol problems are prime candidates for disclosure. What is surprising is the presence of protocol vulnerabilities; one would expect that such problems were discovered and issued during protocol development, specification, and standardization. Their mere existence potentially indicates high protocol complexity.

The vulnerability analysis contained in this paper is, by its nature, static: we have presented a snapshot of known problems with VoIP systems, with no correlation with (and knowledge of) actual attacks exploiting these, or other vulnerabilities. A complete analysis of the threat space would also contain a dynamic component, whereby attacker behavior patterns and trends would be analyzed vis-a-vis actual, deployed VoIP systems or, lacking access to such, simulacra thereof [246].

V. Conclusions

We can draw some preliminary conclusions with respect to threats and focus areas for future research based on the data examined so far. These can be summarized as follows:

1) The large majority of disclosed threats focused on denial of services attacks based on implementation issues. While fault-tolerance techniques can be applied in the case of servers (replication, hot standby, Byzantine fault tolerance, *etc.*), it is less clear how to provide similar levels of protection at acceptable cost and usability to end user devices. Unfortunately, the ease with which mass DoS attacks can be launched over the network against client devices means that they represent an attractive venue for attackers to achieve the same impact.
2) Code injection attacks in their various forms (buffer overflow, cross-site scripting, SQL injection, *etc.*) remain a problem. While a number of techniques have been developed, we need to do a better job at deploying and using them where possible, and devising new techniques suitable for the constrained environments that some vulnerable VoIP devices represent.
3) Weak default configurations remain a problem, as they do across a large class of consumer and enterprise products and software. The situation is likely to be much worse in the real world, considering the complexity of securely configuring a system with as many components as VoIP. Vendors must make an effort to provide secure-by-default configurations, and to educate users how best to protect their systems. Administrators are in need of tools to analyze their existing configurations for vulnerabilities. While some tools that dynamically test network components (*e.g.*, firewalls), we need tools that work higher in the protocol and application stack (*i.e.*, interacting at the user level). Furthermore, we need ways of validating configurations across multiple components and protocols.
4) Finally, there is simply no excuse for protocol-level vulnerabilities. While there exist techniques for analyzing and verifying security protocols, they do not seem to cope well with complexity. Aside from using such tools and continuing their development, protocol designers and standardization committees must consider the impact of their decisions on system implementers, *i.e.*, whether it is likely that a feature or aspect of

the protocol is likely to be misunderstood and/or misimplemented. Simpler protocols are also desirable, but seem incompatible with the trends we have observed in standardization bodies.

Our plans for future work include expanding the data set we used for our analysis to include findings from academic work, adding and presenting more views (classifications) to the data, and developing dynamic views to VoIP-related misbehavior.

ACKNOWLEDGMENTS

This work was supported by the French National Research Agency (ANR) under Contract ANR-08-VERS-017.

REFERENCES

[1] R. Droms, "Dynamic Host Configuration Protocol," RFC 2131 (Draft Standard), Mar. 1997, Updated by RFCs 3396, 4361, 5494.
[2] P.V. Mockapetris, "Domain names - concepts and facilities," RFC 1034 (Standard), Nov. 1987, Updated by RFCs 1101, 1183, 1348, 1876, 1982, 2065, 2181, 2308, 2535, 4033, 4034, 4035, 4343, 4035, 4592.
[3] P.V. Mockapetris, "Domain names - implementation and specification," RFC 1035 (Standard), Nov. 1987, Updated by RFCs 1101, 1183, 1348, 1876, 1982, 1995, 1996, 2065, 2136, 2181, 2137, 2308, 2535, 2845, 3425, 3658, 4033, 4034, 4035, 4343.
[4] K. Sollins, "The TFTP Protocol (Revision 2)," RFC 1350 (Standard), July 1992, Updated by RFCs 1782, 1783, 1784, 1785, 2347, 2348, 2349.
[5] R. Finlayson, "Bootstrap loading using TFTP," RFC 906, June 1984.
[6] P. Srisuresh and K. Egevang, "Traditional IP Network Address Translator (Traditional NAT)," RFC 3022 (Informational), Jan. 2001.
[7] J. Rosenberg, R. Mahy, P. Matthews, and D. Wing, "Session Traversal Utilities for NAT (STUN)," RFC 5389 (Proposed Standard), Oct. 2008.
[8] D. Mills, "Network Time Protocol (Version 3) Specification, Implementation and Analysis," RFC 1305 (Draft Standard), Mar. 1992.
[9] D. Harrington, R. Presuhn, and B. Wijnen, "An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks," RFC 3411 (Standard), Dec. 2002, Updated by RFC 5343.
[10] T. Berners-Lee, R. Fielding, and H. Frystyk, "Hypertext Transfer Protocol – HTTP/1.0," RFC 1945 (Informational), May 1996.
[11] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee, "Hypertext Transfer Protocol – HTTP/1.1," RFC 2616 (Draft Standard), June 1999, Updated by RFC 2817.
[12] T. Dierks and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2," RFC 5246 (Proposed Standard), Aug. 2008.
[13] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications," RFC 3550 (Standard), July 2003, Updated by RFC 5506.
[14] I. Johansson and M. Westerlund, "Support for Reduced-Size Real-Time Transport Control Protocol (RTCP): Opportunities and Consequences," RFC 5506 (Proposed Standard), Apr. 2009.
[15] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler, "SIP: Session Initiation Protocol," RFC 3261 (Proposed Standard), June 2002, Updated by RFCs 3265, 3853, 4320, 4916, 5393.
[16] 3GPP, "Generic Access Network," http://www.3gpp.org/ftp/Specs/html-info/43318.htm, 2009.
[17] J. Franks, P. Hallam-Baker, J. Hostetler, S. Lawrence, P. Leach, A. Luotonen, and L. Stewart, "HTTP Authentication: Basic and Digest Access Authentication," RFC 2617 (Draft Standard), June 1999.
[18] J. Postel, "Transmission Control Protocol," RFC 793 (Standard), Sept. 1981, Updated by RFCs 1122, 3168.
[19] J. Postel, "User Datagram Protocol," RFC 768 (Standard), Aug. 1980.
[20] L. Ong and J. Yoakum, "An Introduction to the Stream Control Transmission Protocol (SCTP)," RFC 3286 (Informational), May 2002.
[21] E. Rescorla and N. Modadugu, "Datagram Transport Layer Security," RFC 4347 (Proposed Standard), Apr. 2006.
[22] M. Handley, E. Rescorla, and IAB, "Internet Denial-of-Service Considerations," RFC 4732 (Informational), Dec. 2006.
[23] M. Handley, V. Jacobson, and C. Perkins, "SDP: Session Description Protocol," RFC 4566 (Proposed Standard), July 2006.
[24] B. Ramsdell, "Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.1 Message Specification," RFC 3851 (Proposed Standard), July 2004.
[25] S. Kent and K. Seo, "Security Architecture for the Internet Protocol," RFC 4301 (Proposed Standard), Dec. 2005.
[26] S. Kent, "IP Encapsulating Security Payload (ESP)," RFC 4303 (Proposed Standard), Dec. 2005.
[27] C. Kaufman, "Internet Key Exchange (IKEv2) Protocol," RFC 4306 (Proposed Standard), Dec. 2005, Updated by RFC 5282.
[28] H. Haverinen and J. Salowey, "Extensible Authentication Protocol Method for Global System for Mobile Communications (GSM) Subscriber Identity Modules (EAP-SIM)," RFC 4186 (Informational), Jan. 2006.
[29] J. Arkko and H. Haverinen, "Extensible Authentication Protocol Method for 3rd Generation Authentication and Key Agreement (EAP-AKA)," RFC 4187 (Informational), Jan. 2006.
[30] J. Arkko, E. Carrara, F. Lindholm, M. Naslund, and K. Norrman, "MIKEY: Multimedia Internet KEYing," RFC 3830 (Proposed Standard), Aug. 2004, Updated by RFC 4738.
[31] Tom Berson, "Skype Security Evaluation," Tech. Rep., October 2005.
[32] S. A. Baset and H. Schulzrinne, "An Analysis of the Skype Peer-to-Peer Telephony Protocol," in *Proceedings of INFOCOM*, April 2006.
[33] P. Biondi and F. Desclaux, "Silver Needle in the Skype," in *BlackHat Europe Conference*, March 2006, www.blackhat.com/presentations/bh-europe-06/bh-eu-06-biondi/bh-eu-06-biondi-up.pdf.
[34] VoIP Security Alliance, "VoIP Security and Privacy Threat Taxonomy, version 1.0," http://www.voipsa.org/Activities/taxonomy.php, October 2005.
[35] The Register, "Two charged with VoIP fraud," http://www.theregister.co.uk/2006/06/08/voip_fraudsters_nabbed/, June 2006.
[36] The Register, "Fugitive VOIP hacker cuffed in Mexico," http://www.theregister.co.uk/2009/02/11/fugitive_voip_hacker_arrested/, February 2009.
[37] "CVE-2007-4753," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-4753, 2007.
[38] "CVE-2007-0431," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-0431, 2007.
[39] "CVE-2007-4553," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-4553, 2007.
[40] "CVE-2003-1114," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2003-1114, 2003.
[41] "CVE-2006-1973," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2006-1973, 2006.
[42] "CVE-2007-0648," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-0648, 2007.
[43] "CVE-2007-2270," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-2270, 2007.
[44] "CVE-2007-4291," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-4291, 2007.
[45] "CVE-2007-4292," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-4292, 2007.
[46] "CVE-2008-3799," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2008-3799, 2008.
[47] "CVE-2008-3800," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2008-3800, 2008.
[48] "CVE-2008-3801," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2008-3801, 2008.
[49] "CVE-2008-3802," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2008-3802, 2008.
[50] "CVE-2009-1158," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2009-1158, 2009.
[51] "CVE-2004-0054," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2004-0054, 2004.
[52] "CVE-2001-0546," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2001-0546, 2001.
[53] "CVE-2002-2266," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2002-2266, 2002.
[54] "CVE-2004-0498," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2004-0498, 2004.
[55] "CVE-2004-2344," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2004-2344, 2004.

[56] "CVE-2004-2629," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2004-2629, 2004.
[57] "CVE-2004-2758," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2004-2758, 2004.
[58] "CVE-2007-4429," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-4429, 2007.
[59] "CVE-2006-5084," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2006-5084, 2006.
[60] "CVE-2005-3267," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2005-3267, 2005.
[61] "CVE-2004-1777," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2004-1777, 2004.
[62] "CVE-2003-1108," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2003-1108, 2003.
[63] "CVE-2003-1109," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2003-1109, 2003.
[64] "CVE-2003-1110," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2003-1110, 2003.
[65] "CVE-2003-1111," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2003-1111, 2003.
[66] "CVE-2003-1112," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2003-1112, 2003.
[67] "CVE-2003-1113," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2003-1113, 2003.
[68] "CVE-2003-1115," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2003-1115, 2003.
[69] "CVE-2004-0504," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2004-0504, 2004.
[70] "CVE-2005-4466," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2005-4466, 2005.
[71] "CVE-2006-5445," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2006-5445, 2006.
[72] "CVE-2006-2924," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2006-2924, 2006.
[73] "CVE-2006-0739," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2006-0739, 2006.
[74] "CVE-2006-0738," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2006-0738, 2006.
[75] "CVE-2006-0737," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2006-0737, 2006.
[76] "CVE-2007-6371," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-6371, 2007.
[77] "CVE-2007-5583," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-5583, 2007.
[78] "CVE-2007-5537," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-5537, 2007.
[79] "CVE-2007-4924," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-4924, 2007.
[80] "CVE-2007-4459," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-4459, 2007.
[81] "CVE-2007-4455," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-4455, 2007.
[82] "CVE-2007-4382," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-4382, 2007.
[83] "CVE-2007-4366," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-4366, 2007.
[84] "CVE-2007-3445," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-3445, 2007.
[85] "CVE-2007-3444," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-3444, 2007.
[86] "CVE-2007-3443," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-3443, 2007.
[87] "CVE-2007-3442," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-3442, 2007.
[88] "CVE-2007-3441," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-3441, 2007.
[89] "CVE-2007-3437," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-3437, 2007.
[90] "CVE-2007-3369," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-3369, 2007.
[91] "CVE-2007-3368," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-3368, 2007.
[92] "CVE-2007-3363," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-3363, 2007.
[93] "CVE-2007-3362," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-3362, 2007.
[94] "CVE-2007-3361," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-3361, 2007.
[95] "CVE-2007-3351," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-3351, 2007.
[96] "CVE-2007-3350," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-3350, 2007.
[97] "CVE-2007-3349," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-3349, 2007.
[98] "CVE-2007-3348," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-3348, 2007.
[99] "CVE-2007-3322," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-3322, 2007.
[100] "CVE-2007-3321," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-3321, 2007.
[101] "CVE-2007-3318," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-3318, 2007.
[102] "CVE-2007-3317," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-3317, 2007.
[103] "CVE-2007-2297," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-2297, 2007.
[104] "CVE-2007-1693," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-1693, 2007.
[105] "CVE-2007-1650," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-1650, 2007.
[106] "CVE-2007-1594," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-1594, 2007.
[107] "CVE-2007-1590," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-1590, 2007.
[108] "CVE-2007-1561," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-1561, 2007.
[109] "CVE-2007-1542," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-1542, 2007.
[110] "CVE-2007-1306," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-1306, 2007.
[111] "CVE-2007-0961," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-0961, 2007.
[112] "CVE-2008-0095," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2008-0095, 2008.
[113] "CVE-2008-0263," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2008-0263, 2008.
[114] "CVE-2008-1249," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2008-1249, 2008.
[115] "CVE-2008-1741," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2008-1741, 2008.
[116] "CVE-2008-1745," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2008-1745, 2008.
[117] "CVE-2008-1747," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2008-1747, 2008.
[118] "CVE-2008-1748," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2008-1748, 2008.
[119] "CVE-2008-1959," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2008-1959, 2008.
[120] "CVE-2008-2119," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2008-2119, 2008.
[121] "CVE-2008-2732," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2008-2732, 2008.
[122] "CVE-2008-2733," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2008-2733, 2008.
[123] "CVE-2008-2734," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2008-2734, 2008.
[124] "CVE-2008-2735," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2008-2735, 2008.
[125] "CVE-2008-3157," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2008-3157, 2008.
[126] "CVE-2008-3210," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2008-3210, 2008.
[127] "CVE-2008-3778," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2008-3778, 2008.
[128] "CVE-2008-4444," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2008-4444, 2008.
[129] "CVE-2008-5180," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2008-5180, 2008.

[130] "CVE-2008-6140," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2008-6140, 2008.
[131] "CVE-2008-6574," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2008-6574, 2008.
[132] "CVE-2008-6575," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2008-6575, 2008.
[133] "CVE-2009-0871," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2009-0871, 2009.
[134] "CVE-2009-0636," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2009-0636, 2009.
[135] "CVE-2009-0630," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2009-0630, 2009.
[136] "CVE-2009-0631," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2009-0631, 2009.
[137] "CVE-2007-5591," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-5591, 2007.
[138] "CVE-2007-5556," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-5556, 2007.
[139] "CVE-2007-5369," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-5369, 2007.
[140] "CVE-2007-2886," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-2886, 2007.
[141] "CVE-2006-7121," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2006-7121, 2006.
[142] "CVE-2006-6411," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2006-6411, 2006.
[143] "CVE-2006-5233," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2006-5233, 2006.
[144] "CVE-2006-5231," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2006-5231, 2006.
[145] "CVE-2005-3989," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2005-3989, 2005.
[146] "CVE-2004-1977," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2004-1977, 2004.
[147] "CVE-2002-0882," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2002-0882, 2002.
[148] "CVE-2002-0880," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2002-0880, 2002.
[149] "CVE-2002-0835," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2002-0835, 2002.
[150] "CVE-2007-3436," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-3436, 2007.
[151] "CVE-2005-4464," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2005-4464, 2005.
[152] "CVE-2007-5488," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-5488, 2007.
[153] "CVE-2007-5411," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-5411, 2007.
[154] "CVE-2008-0582," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2008-0582, 2008.
[155] "CVE-2008-0583," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2008-0583, 2008.
[156] "CVE-2008-0454," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2008-0454, 2008.
[157] "CVE-2006-2925," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2006-2925, 2006.
[158] "CVE-2007-2191," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-2191, 2007.
[159] "CVE-2008-1251," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2008-1251, 2008.
[160] "CVE-2008-6509," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2008-6509, 2008.
[161] "CVE-2008-6573," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2008-6573, 2008.
[162] "CVE-1999-0938," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0938, 1999.
[163] "CVE-2008-1250," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2008-1250, 2008.
[164] "CVE-2008-6708," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2008-6708, 2008.
[165] "CVE-2007-4498," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-4498, 2007.
[166] "CVE-2007-3319," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-3319, 2007.
[167] "CVE-2007-3177," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-3177, 2007.
[168] "CVE-2007-0334," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-0334, 2007.
[169] "CVE-2005-4050," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2005-4050, 2005.
[170] "CVE-2007-4294," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-4294, 2007.
[171] "CVE-2007-4295," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-4295, 2007.
[172] "CVE-2004-0056," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2004-0056, 2004.
[173] "CVE-2004-0117," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2004-0117, 2004.
[174] "CVE-2005-3265," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2005-3265, 2005.
[175] "CVE-2004-1114," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2004-1114, 2004.
[176] "CVE-2003-0761," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2003-0761, 2003.
[177] "CVE-2006-4029," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2006-4029, 2006.
[178] "CVE-2006-3594," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2006-3594, 2006.
[179] "CVE-2006-3524," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2006-3524, 2006.
[180] "CVE-2006-0359," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2006-0359, 2006.
[181] "CVE-2006-0189," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2006-0189, 2006.
[182] "CVE-2007-5788," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-5788, 2007.
[183] "CVE-2007-3438," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-3438, 2007.
[184] "CVE-2007-2293," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-2293, 2007.
[185] "CVE-2007-0746," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-0746, 2007.
[186] "CVE-2008-0528," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2008-0528, 2008.
[187] "CVE-2008-0530," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2008-0530, 2008.
[188] "CVE-2008-0531," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2008-0531, 2008.
[189] "CVE-2008-2085," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2008-2085, 2008.
[190] "CVE-2007-4489," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-4489, 2007.
[191] "CVE-2005-2081," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2005-2081, 2005.
[192] "CVE-2003-0819," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2003-0819, 2003.
[193] "CVE-2005-1461," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2005-1461, 2005.
[194] "CVE-2008-2545," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2008-2545, 2008.
[195] "CVE-2008-1805," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2008-1805, 2008.
[196] "CVE-2007-5989," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-5989, 2007.
[197] "CVE-2007-3896," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-3896, 2007.
[198] "CVE-2008-6709," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2008-6709, 2008.
[199] "CVE-2008-1334," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2008-1334, 2008.
[200] "CVE-2006-2312," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2006-2312, 2006.
[201] "CVE-2005-2181," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2005-2181, 2005.
[202] "CVE-2005-2182," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2005-2182, 2005.
[203] "CVE-2007-5791," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-5791, 2007.

[204] "CVE-2007-3347," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-3347, 2007.
[205] "CVE-2007-3320," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-3320, 2007.
[206] "CVE-2006-0305," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2006-0305, 2006.
[207] "CVE-2006-0302," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2006-0302, 2006.
[208] "CVE-2005-3804," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2005-3804, 2005.
[209] "CVE-2005-3724," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2005-3724, 2005.
[210] "CVE-2005-3715," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2005-3715, 2005.
[211] "CVE-2006-0360," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2006-0360, 2006.
[212] "CVE-2007-3439," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-3439, 2007.
[213] "CVE-2006-0374," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2006-0374, 2006.
[214] "CVE-2005-3723," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2005-3723, 2005.
[215] "CVE-2005-3721," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2005-3721, 2005.
[216] "CVE-2005-3718," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2005-3718, 2005.
[217] "CVE-2007-3440," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-3440, 2007.
[218] "CVE-2008-1248," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2008-1248, 2008.
[219] "CVE-2006-5038," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2006-5038, 2006.
[220] "CVE-2008-4874," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2008-4874, 2008.
[221] "CVE-2007-3047," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-3047, 2007.
[222] "CVE-2006-0834," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2006-0834, 2006.
[223] "CVE-2005-3803," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2005-3803, 2005.
[224] "CVE-2005-3719," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2005-3719, 2005.
[225] "CVE-2005-3717," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2005-3717, 2005.
[226] "CVE-2005-3716," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2005-3716, 2005.
[227] "CVE-2005-0745," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2005-0745, 2005.
[228] "CVE-2002-0881," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2002-0881, 2002.
[229] "CVE-2006-0375," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2006-0375, 2006.
[230] "CVE-2005-3725," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2005-3725, 2005.
[231] "CVE-2007-5361," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-5361, 2007.
[232] "CVE-2008-5871," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2008-5871, 2008.
[233] "CVE-2005-3722," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2005-3722, 2005.
[234] "CVE-2008-4875," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2008-4875, 2008.
[235] "CVE-2008-6706," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2008-6706, 2008.
[236] "CVE-2008-6707," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2008-6707, 2008.
[237] "CVE-2007-6095," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-6095, 2007.
[238] "CVE-2008-1114," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2008-1114, 2008.
[239] "CVE-2008-1113," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2008-1113, 2008.
[240] "CVE-2002-1935," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2002-1935, 2002.
[241] "CVE-2006-4032," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2006-4032, 2006.
[242] "CVE-2008-3903," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2008-3903, 2008.
[243] "CVE-2007-5469," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-5469, 2007.
[244] "CVE-2007-5468," http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-5468, 2007.
[245] R. State, O. Festor, H. Abdelanur, V. Pascual, J. Kuthan, R. Coeffic, J. Janak, and J. Floroiu, "SIP digest authentication relay attack," draft-state-sip-relay-attack-00, Mar. 2009.
[246] M. Nassar, R. State, and O. Festor, "VoIP Honeypot Architecture," in *Proceedings of the* 10^{th} *IFIP/IEEE International Symposium on Integrated Network Management*, May 2007, pp. 109–118.